Development of an EMG based SVM supported control solution for the PlatypOUs education mobile robot using MindRove headset

Erick Noboa *,****, **Melinda Rácz** **, **László Szűcs** *, **Péter Galambos** *, **Gergely Márton** **,***, and **György Eigner** ****

* *Bejczy Antal Center for Intelligent Robotics, Robotics Special College, Óbuda University, Budapest, Hungary (e-mail: erick7327@gmail.com, laszlo.szucs@irob.uni-obuda.hu, peter.galambos@irob.uni-obuda.hu).*
** *Integrative Neuroscience Research Group, Institute of Cognitive Neuroscience and Psychology, Research Centre for Natural Sciences, Budapest, Hungary (e-mail: racz.melinda.9157@gmail.com)*
*** *MindRove Kft., Mészáros u. 60/C. H-1016 Budapest, Hungary (e-mail: gergely.marton@mindrove.com)*
**** *Physiological Controls Research Center, Óbuda University, Budapest, Hungary (e-mail: eigner.gyorgy@uni-obuda.hu)*

Abstract: This paper describes the development of PlatypOUs – an open-source electromyography (EMG)-controlled mobile robot platform that uses the MindRove Brain Computer Interface (BCI) headset as signal acquisition unit, implementing remote control. Simultaneously with the physical mobile robot, simulation environment is also prepared using Gazebo, within the Robot Operating System (ROS) framework, with the same capabilities as the physical device, from the point of view of the ROS. The purpose of the PlatypOUs project is to create a tool for STEM-based education, and it involves two major disciplines: mobile robotics and machine learning, with several sub-areas included in each. The use of the platform and the simulation environment exposes students to hands-on laboratory sessions, which contribute to their progression as engineers. An important feature of our project is that the platform is made up of open-source and easily available commercial hardware and software components. In this paper, an electromyography (EMG) based controller has been developed using support vector machine (SVM) based classification for robot control purposes.

Keywords: Mobile robots, Intelligent controllers, Intelligent interfaces, Human operator support, Artificial intelligence, Robotics

1. INTRODUCTION

In STEM education (especially at higher levels) laboratory demonstration tools are important in order to provide the students with an overview of the available technologies in a specific domain. In particular, mobile robotics is one of the most comprehensive instruments for educators since these are excellent for technological demonstrations. Many examples can be found in the literature where mobile robot platforms are applied for educational purposes, e.g. Lalonde et al. (2006), Ali (2011), Arvin et al. (2019).

Despite the platforms being already available on the market, for laboratories of university departments and special colleges own prototypes are worth building, since the development process can bring the researchers' and students' communities together, and making of an open-source platform can catalyze the advancement of students from many areas. In these days, the big engineering challenges can only be solved by working in interdisciplinary teams, which is especially true in robotics Lalonde et al. (2006).

An important aspect of this is how to draw attention of students to the specific topics. In this work, we introduce the novel mobile robot platform of Bejczy Antal Center of Intelligent Robotics of Óbuda University and the related Robotic Special College—PlatypOUs—which is designed for educational purposes and is equipped with up-to-date sensors and technology.

⋆ Project no. 2019-1.3.1-KK-2019-00007. has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2019-1.3.1-KK funding scheme. Gergely Márton is thankful to the Hungarian National Research, Development and Innovation Office (FK132823 grant). This research has been partially funded by the National Brain Research Program (2017-1.2.1-NKP-2017-00002). Gy. Eigner and P. Gamabos were supported by the Eötvös Loránd Research Network Secretariat under grant agreement no. ELKH KÖ-40/2020 ('Development of cyber-medical systems based on AI and hybrid cloud methods').

Our original aim was to develop a machine learning based control solution employing a known BCI (MindRove) using EEG signals. Although EEG headsets are widely used as non-invasive interfaces for many purposes from control to gaming Jackson and Mappus (2010), using EEG signals for control purposes is not a trivial task and requires deep understanding of both disciplines Meng et al. (2016). This article presents the first step we took in this plan, where we introduce the developed environments and an EMG-based controller, which was selected due to multiple reasons. In our further work, our goal will be to develop an interface that utilizes the EEG signals as well. On this roadmap, one of the biggest challenges is how the signals of the BCI can be interpreted and utilized. A possible solution is the machine learning-based signal classification, which is also widely used for feature detection in EMG and EEG signals Meng et al. (2016).

Our control interface was intended to be accessible and easy-to-use for every student during laboratory practices. Unfortunately, in case of any type of EEG-BCIs, there is a minority (approximately the 20 %) of users that are unable to operate it for some reason, even though they are healthy—this phenomenon is commonly referred to as BCI illiteracy Allison and Neuper (2010). Therefore, we abandoned the concept of developing a purely EEG-BCI and shifted to the implementation of a more easily applicable EMG-based system that uses signals which are considered artifacts in most of the cases, for instance blinking or raising eyebrows, although, the idea of utilizing an EEG measurement device for signal acquisition purposes was kept.

The paper is structured as follows. First, we introduce the involved technologies and frameworks, the robot platform and the applied classifier. Then, we show the results and evaluate them. Finally, we conclude our work.

2. MATERIALS AND METHODS

2.1 Involved Technologies

Our goal was to develop an EMG-based controller for our mobile robot platform that required an artificial intelligence (AI) based interpreter which can translate the signal to commands our robot can understand understand. We applied MindRove BCI headset where we would be able to access a rich data content through the SDK of the platform. That allows the implementation of a support vector machine (SVM) Cortes and Vapnik (1995). We considered multiple classifier opportunities, however, according to previous studies the usage of SVM provides sufficient solution Rejani and Selvi (2009). We created an SVM that can be trained to recognize two different signals and use them to provide the user with two control commands from the data obtained with the MindRove headset. The first command allows to switch between directions, while the second command gives the instruction to move the robot in the selected direction. The connection between the headset and the robotic platform was realized using the MQTT communication protocol Banks et al. (2019). For communication purposes we created a dedicated Local Area Network (LAN) and a MQTT broker, which are required to handle the asynchronous information trade Sahadevan et al. (2017) between the machine that is processing EMG signals and the ROS node running in the robot's computer which gives velocity instructions to drive each wheel independently. In this case, *Mosquitto* MQTT broker is implemented as a broker to manage the communication between the machines involved Light (2017).

2.2 The PlatypOUs Robot Platform

The PlatypOUs robot is a differential drive mobile platform with two wheels on the front and a caster wheel on the back. The two driven wheels have brushless DC hub motors, which can be independently controlled by an ODrive motor driver board ODrive (2013). This board is connected via USB-Serial interface to the main computer, which is an Intel NUC8i5BEH mini PC, running Ubuntu 20.04. The control system is based on the open-source robotics middleware ROS, the version used is Noetic, currently the newest release Quigley et al. (2009). The environment along with all necessary dependencies is built into a Docker image and is running inside a container Merkel (2014), White and Christensen (2017). A ROS node was developed to control the ODrive board from the ROS environment. The node listens for ROS messages called Twist, which contain the desired linear and angular velocities for the platform. From the message data it calculates the appropriate speeds for the motors and sends them to the motor driver board, which uses encoder feedback, and a PID control loop, to make the platform move in the desired way.

The robot can also be used in the Gazebo simulator Koenig and Howard (2004). The simulation was created to match the real robot as closely as possible, both in ROS interface and in physical behavior. Thanks to this, most things developed and tested in the simulated environment can be moved to the real hardware easily, with minimal changes. This is especially useful in the developing and testing of control methods, because it is faster, easier and more repeatable than doing the same on the real hardware.

To integrate the data flowing from the headset to the ROS environment, a node was made to listen for MQTT messages. The robot is connected to a wireless LAN network, and the node connects to the MQTT broker using this connection. Received messages contain the information needed to control the wheels. This can be one of 4 directions, or a stop command, represented by numbers. Based on the number, a ROS Twist message is sent, containing the corresponding linear and angular velocities. These are predefined constant values, selected as 0.2 m/s for linear, and 0.5 rad/s for angular velocities, as these values were found to be useful during testing. A safety timeout was implemented in the motor driver node, which stops the wheels if a velocity command is not received for a selected amount of time. For the control using the headset, a timeout of 1 second was used.

2.3 Data acquisition and Pre-processing

Hardware As data acquisition device, a MindRove BCI headset was applied (shown in Figure 1). The MindRove BCI headset is a commercial, WiFi-based wireless six-channel EEG headset with semi-dry electrodes Min-

Fig. 1. The MindRove EEG headset.

dRove (2021). Data was acquired at a sampling rate of $f_s = 500$ Hz in every channels.

Data acquisition paradigm A simple Windows Presentation Foundation (WPF) application based on the public SDK provided by MindRove was implemented to acquire training samples for the classifier, and the same application has been applied for control actions issued by the user. Due to the sensitivity of BCIs to electrode placement, we decided that the classifier should be retrained on each program execution.

The application needs two parameters upon start, namely, the number of the training samples per command type and the length of the samples (n_s, in seconds). The user interface of the app is shown on Figure 2. After the parameters got entered, the user is prompted to perform the preferred commands cued by the application. The trials are in a random order.

Data preprocessing Training samples and control commands are formatted identically: each sample is based on a time window—an array with $n_s \cdot f_s$ rows and $n_{ch} = 6$ columns. Samples are produced with a frequency of $1/n_s$, thus subsequent samples do not overlap in time.

These samples get fed into a discrete-time Fourier transform (for this purpose, we used Emgu CV—a C# wrapper for OpenCV EMGU (2016) then the power spectrum of the sample is taken, restricted to the components in the range of $(0, 80]$ Hz. This domain encompasses frequency bands from delta to beta and includes the majority of the gamma band.

Finally, the samples are normalized one-by-one using the standardization formula below:

$$x'_i = \frac{x_i - \mu_i}{\sigma_i}, \tag{1}$$

where x'_i and x_i denote the normalized and original samples, μ_i and σ_i the average and standard deviation of the original sample, respectively.

2.4 SVM-based classifier

For the purpose of classification, we used the SVM implementation provided by Emgu CV EMGU (2016). This implementation is capable of choosing the optimal SVM type and parameter set for the classifier during training.

After training, movement direction and status (start/stop) are displayed in MindRove SVM Controller as shown in Figure 3.

2.5 Finalized EEG-driven Architecture

In the previous section we have mentioned that the result obtained from the SVM classifier are three integer numbers, as Figure 4 shows, where a value of 1 represents a *"switch direction instruction"*. In order to keep track of this command, an extra variable was involved to store the actual direction and jump to the next position every time that a *change* direction instruction was detected by the classifier as the flow in the Figure 5 shows. The code switch direction in a clockwise order, starting in the *"Forward"* position as the Figure 8 shows in the results section.

On the other hand, a ROS node, that runs in the computer inside of the mobile platform, manages the information obtained from the classifier through the *MQTT "/test" topic,* and builds a *"twist" type message* with the desired velocity values which the node publishes in the ros-topic

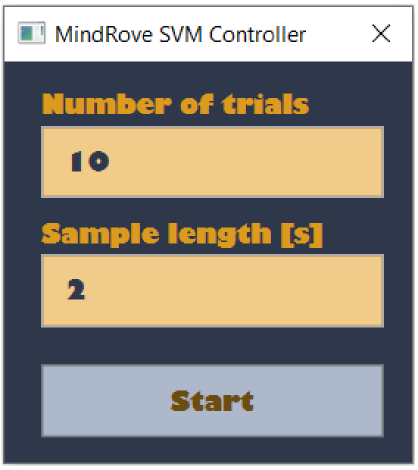

Fig. 2. User interface of MindRove SVM Controller with the default input parameters.

Fig. 3. Moving in the forward direction displayed by MindRove SVM Controller.

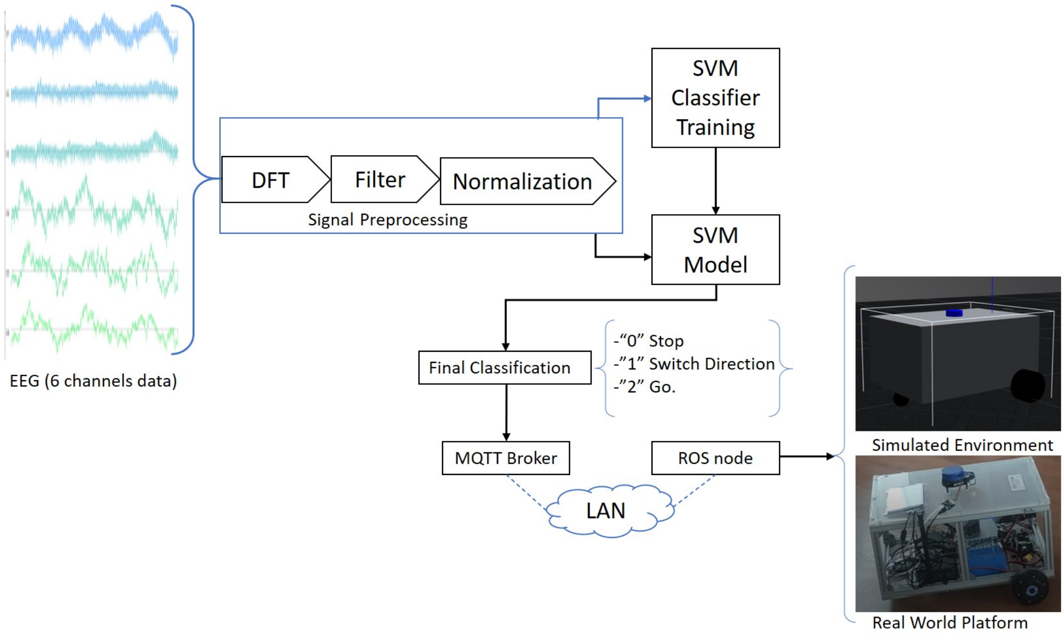

Fig. 4. Signal acquisition and classification flow.

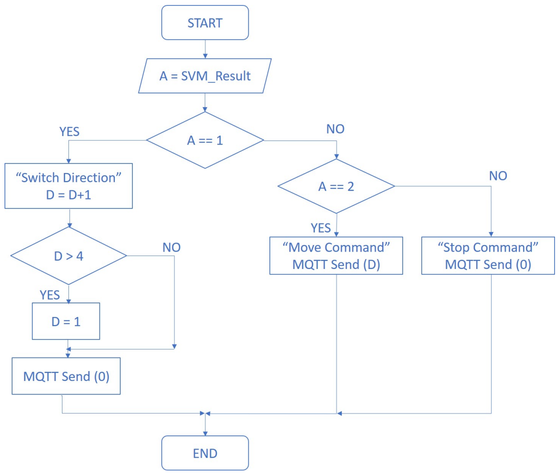

Fig. 5. Flow Diagram of the Direction Tracking process.

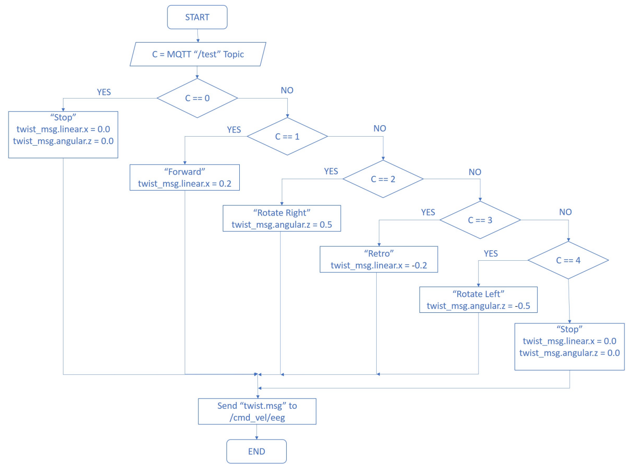

Fig. 6. Flow Diagram of the velocities assignation in the ROS node.

"/cmd vel/eeg" as the Figure 6 shows. This message is processed by the motor-driver node, which actuates the desired motion.

3. RESULTS

3.1 SVM classification and assessment

The pre-trained SVM model got evaluated after the training process. For this purpose, an array of 20 labeled samples per command was acquired using the same setup we installed for classifier training:

- the headset was placed above the parietal cortex of the user; this placement proved to be appropriate to collect the artifacts necessary for the control;
- as control commands, we used *"eyebrow raising"* (*"change direction"*) and *"Strong wink"* (*"move"*).

The samples got pre-processed identically to the training data. Then the test data was fed into the pre-trained classifier for obtaining the reconstruction of the labels.

A confusion matrix calculated using results from a session performed by an experienced user is displayed in Figure 7. By choosing this control command set, we could achieve an accuracy of 86.667 %.

3.2 Robot control tests - real/virtual

The complete system was tested inside the iRob laboratory environment obtaining the following results. Figure 8 demonstrates the classification process where an action of *"change direction"* is performed, in this case a *"raising eyebrows"* action has been chosen—it can be seen in the center of the bottom row of Figure 8. Here we can notice that the platform remains in the same position, while the selected direction has changed taking a clockwise step, because *"move"* command has not been sent yet.

	0	1	2
0	13	7	0
1	0	19	1
2	0	0	20

Fig. 7. Confusion matrix showing the command types predicted by the classifier. The rows correspond to the real labels, the columns the predicted ones. 0, 1, 2 denotes idle state, eyebrow raising and chewing, respectively.

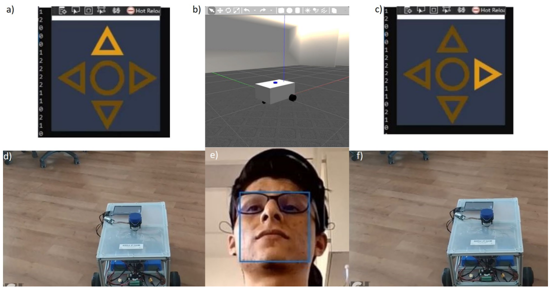

Fig. 8. Previous (figures a & d) and following (figures c & f) state of the real and simulated (figure b) PlatypOUs platform when the user performs a change direction artifact raising eyebrows (figure e).

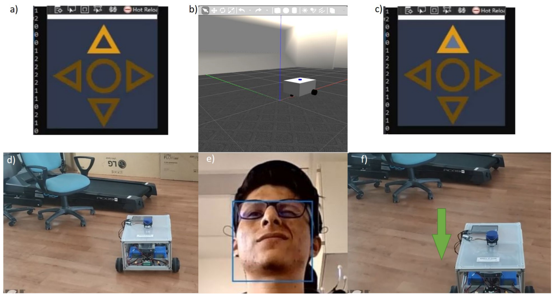

Fig. 9. Previous (figures a & d) and following (figures c & f) state of the real and simulated (figure b) PlatypOUs platform performing a linear when the user performs a move artifact winking (figure e).

In the Figures 9 and 10 the classification results of *"move"* actions can be seen, for this purpose a *"strong wink"* movement has been applied. In this pictures, it can be noticed that the PlatypOUs platform is performing a *forward* and *rotation* movement, respectively.

All these figures show the result of the classification process and the movements executed by the mobile platform in the simulated Gazebo environment. As can be observed in the top center field of each figure (b), the robot has performed the same moves than the platform in real life. To evaluate the performance of the complete system, a sequence of movements was previously defined to be followed by the user during tests, this sequence was designed to observe the behave of the classifier along with the response of the mobile platform involving both commands to choose the aspired direction and effectuate the motion. The selected pattern of movements goes as follows: *Rotation Right, Linear Forward, Rotation Left, Linear Forward, Rotation Left, linear Backward, Rotation Right.* Where the system showed a precision of 84.62 % which agrees with the evaluation of the model in the previous section. Another important characteristic observed during the test was the time of response of the system, starting when the user performed an specific artifact command until the time when the platform executed the movement an average interval equal to 1.12 (s) was noticed, which can be attributed as the required time to process the data.

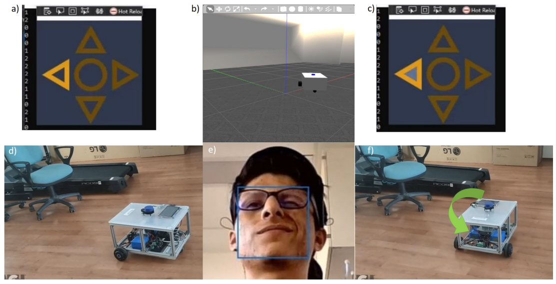

Fig. 10. Previous (figures a & d) and following (figures c & f) state of the real and simulated (figure b) PlatypOUs platform performing a rotational movement when the user performs a move artifact winking (figure e).

4. DISCUSSION

In the presented approach, we applied an SVM classifier to determine two different EMG-based face artifacts from MindRove BCI signals, e.g. moving eyebrows, blinking or chewing, in order to give change direction and move commands to the mobile platform -PlatypOUs-. The accuracy of the obtained classifier has been evaluated by the application of 20 samples per command into the pre-trained model, the first thing we have noticed here is that the accuracy of each model can change drastically from one training process to another, this because of the variability of the contact position between the electrodes of the headset and the user's head, this is why we have decided to train the classifier every time the paradigm runs. Even considering this limitation the user was able to obtain an accuracy of 86.667 % which is acceptable to perform a stable control of the platform.

In the other hand, we have probed that a team of students with different knowledge, fields and skills can successfully develop a platform for STEM-based education purposes which will help to improve a plenty of abilities to the hole student community in the future, allowing them to practice all the theory shared in the classroom by professors in different knowledge areas such as mathematics, control theory, electronics, CAD/CAM/CAE, software development, etc. Preparing them for the world of work in a realistic way.

5. CONCLUSION

Using open-source and easily available components, a new robot platform has been designed for educational purposes and presented herein in depth along with its mechanical and functional components. The technologies the platform based on are are diverse enough to require expertise in several areas. One of the most challenging tasks was the data acquisition and pre-processing of the EMG signal.

The platform gives new horizons to STEM as it is a custom and modular design. The cost of PlatypOUs is around 1200 € (as of 2021 Q2, incorporates Intel RealSense D435i RGBD camera). The platform currently has basic navigational and mobility equipment. In the future, the robot platform will be improved by adding new software features based on new sensors and actuators on both the physical and the simulation side. All of the design and implementation was—and will be—performed by students.

ACKNOWLEDGEMENTS

We would like to thank the following team members of PlatypOUs project for all the committed work and time they spent improving and testing the platform: Borsa Détár, Panna Zsoldos, Ema Basovic, Aleksandar Pencik and L. Szűcs of the Applied Informatics and Applied Mathematics Doctoral School of Óbuda University who also supported the project.

REFERENCES

Ali, W.G. (2011). A semi-autonomous mobile robot for education and research. *Journal of King Saud University - Engineering Sciences*, 23, 131–138.

Allison, B.Z. and Neuper, C. (2010). *Brain-Computer Interfaces*, chapter Could Anyone Use a BCI?, 35–54. Springer-Verlag London Limited.

Arvin, F., Espinosa, J., Bird, B., West, A., Watson, S., and Lennox, B. (2019). Mona: an Affordable Open-Source Mobile Robot for Education and Research. *Journal of Intelligent & Robotic Systems*, 94, 761–775.

Banks, A., Briggs, E., Borgendale, K., and Gupta, R. (2019). MQTT Version 5.0. OASIS Standard, `https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html`. Latest version: `https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html`.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273–297.

EMGU (2016). EMGU CV. `https://www.emgu.com/wiki/index.php/Main_Page`. Accessed: 2021-04-26.

Jackson, M.M. and Mappus, R. (2010). *Brain-Computer Interfaces*, chapter Applications for Brain-Computer Interfaces, 89–103. Springer-Verlag London Limited.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, 2149–2154. IEEE, Sendai, Japan. doi:10.1109/IROS.2004.1389727. URL `http://ieeexplore.ieee.org/document/1389727/`.

Lalonde, J.F., Bartley, C.P., and Nourbakhsh, I. (2006). Mobile robot programming in education. *Proceedings 2006 IEEE International Conference on Robotics and Automation*, 2006, 345–350.

Light, R. (2017). Mosquitto: server and client implementation of the MQTT protocol. *The Journal of Open Source Software*, 2(13), 256.

Meng, J., Zhang, S., Bekyo, A., Olsoe, J., Baxter, B., and He, B. (2016). Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks. *Scientific Reports*, 6, 38565.

Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239), 2.

MindRove (2021). MindRove. `http://mindrove.com`. Accessed: 2021-04-26.

ODrive (2013). ODrive. `https://odriverobotics.com/`. Accessed: 2021-04-26.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. (2009). ROS: an open-source Robot Operating System. 3. Journal Abbreviation: ICRA Workshop on Open Source Software Publication Title: ICRA Workshop on Open Source Software.

Rejani, Y.I.A. and Selvi, S.T. (2009). Early Detection of Breast Cancer using SVM Classifier Technique. *International Journal on Computer Science and Engineering*, 1(3), 127–130.

Sahadevan, A., Mathew, D., Mookathana, J., and Jose, B.A. (2017). An Offline Online Strategy for IoT Using MQTT. *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, 369–373.

White, R. and Christensen, H. (2017). ROS and Docker. In A. Koubaa (ed.), *Robot Operating System (ROS)*, volume 707, 285–307. Springer International Publishing, Cham. doi:10.1007/978-3-319-54927-9_9. URL `http://link.springer.com/10.1007/978-3-319-54927-9_9`. Series Title: Studies in Computational Intelligence.